Package 'whitening'

January 12, 2019

Version 1.1.1

Date 2019-01-12

Title Whitening and High-Dimensional Canonical Correlation Analysis

Author Korbinian Strimmer, Takoua Jendoubi, Agnan Kessy, Alex Lewin

Maintainer Korbinian Strimmer <strimmerlab@gmail.com>

Depends R (>= 3.0.2), corpcor (>= 1.6.9)

Imports stats

Suggests

Description Implements the whitening methods (ZCA, PCA, Cholesky, ZCA-cor, and PCA-cor) discussed in Kessy, Lewin, and Strimmer (2018) ``Optimal whitening and decorrelation", <doi:10.1080/00031305.2016.1277159>, as well as the whitening approach to canonical correlation analysis allowing negative canonical correlations described in Jendoubi and Strimmer (2019) ``A whitening approach to probabilistic canonical correlation analysis for omics data integration", <doi:10.1186/s12859-018-2572-9>.

License GPL (>= 3)

URL http://strimmerlab.org/software/whitening/

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-12 17:10:03 UTC

R topics documented:

whitening-package *The whitening Package*

Description

The "whitening" package implements the whitening methods (ZCA, PCA, Cholesky, ZCA-cor, and PCA-cor) discussed in Kessy, Lewin, and Strimmer (2018) as well as the whitening approach to canonical correlation analysis allowing negative canonical correlations described in Jendoubi and Strimmer (2019).

Author(s)

Korbinian Strimmer (http://strimmerlab.org/) with Takoua Jendoubi, Agnan Kessy, and Alex Lewin.

References

Kessy, A., A. Lewin, and K. Strimmer. 2018. Optimal whitening and decorrelation. The American Statistician. 72: 309-314. https://doi.org/10.1080/00031305.2016.1277159

Jendoubi, T., and K. Strimmer 2019. A whitening approach to probabilistic canonical correlation analysis for omics data integration. BMC Bioinformatics 20: 15. https://doi.org/10.1186/s12859-018-2572-9

Website: http://strimmerlab.org/software/whitening/

See Also

whiteningMatrix. whiten, cca, and scca.

corplot *Plots of Correlations and Loadings*

Description

corplot computes the correlation within and between X and Y and displays the three corresponding matrices visusally.

loadplot computes the squared loadings for X and Y and plots the resulting matrices.

Usage

```
corplot(cca.out, X, Y)
loadplot(cca.out, numScores)
```

Arguments

cca.out	output from the scca or cca function.
X, Y	input data matrices.
numScores	number of CCA scores shown in plot.

Value

A plot.

Author(s)

Korbinian Strimmer (http://strimmerlab.org).

Part of the plot code was adapted from the img.matcor function in the CCA package and from the image.plot function in the fields package.

See Also

scca.

---

forina1986 *Forina 1986 Wine Data - Extended UCI Wine Data*

Description

The forina1986 dataset describes 27 properties of 178 samples of wine from three grape varieties (59 Barolo, 71 Grignolino, 48 Barbera) as reported in Forina et al. (1986).

Usage

```
data(forina1986)
```

Format

A list containing the following components:

attribs collects measurements for 27 attributes of 178 wine samples.

type describes the variety ("Barolo", "Grignolino", or "Barbera").

Details

This data set contains the full set of covariates described in Forina et al. (1986) except for Sulphate (variable 15 in Forina et al. 1986). These are: 1) Alcohol, 2) Sugar-free extract, 3) Fixed acidity, 4) Tartaric acid, 5) Malic acid, 6) Uronic acids, 7) pH, 8) Ash, 9) Alcalinity of ash, 10) Potassium, 11) Calcium, 12) Magnesium, 13) Phosphate, 14) Chloride, 15) Total phenols, 16) Flavanoids, 17) Nonflavanoid phenols, 18) Proanthocyanins, 19) Color intensity, 20) Hue, 21) OD280/OD315 of diluted wines, 22) OD280/OD315 of flavonoids, 23) Glycerol, 24) 2-3-butanediol, 25) Total nitrogen, 26) Proline, and 27) Methanol.

The UCI wine data set (https://archive.ics.uci.edu/ml/datasets/wine) is a subset of the Forina et al. (1986) data set comprising only 13 variables.

Source

The original data matrix is available from https://www.researchgate.net/profile/Michele_Forina/publication/271908647_Wines_MForina_CArmanino_MCastino_MUbigli_Multivariate_data_analysis_as_discriminating_method_of_the_origin_of_wines_Vitis_25_189-201_1986/.

References

Forina, M., Armanino, C., Castino, M., and Ubigli, M. Multivariate data analysis as a discriminating method of the origin of wines. Vitis 25:189-201 (1986). https://ojs.openagrar.de/index.php/VITIS/article/view/5950.

Examples

```
# load whitening library
library("whitening")

# load Forina 1986 wine data set
data(forina1986)

table(forina1986$type)
#    Barolo Grignolino    Barbera
#        59         71         48

dim(forina1986$attrib)
# 178  27

colnames(forina1986$attrib)
# [1] "Alcohol"                      "Sugar-free extract"
# [3] "Fixed acidity"                "Tartaric acid"
# [5] "Malic acid"                   "Uronic acids"
# [7] "pH"                           "Ash"
# [9] "Alkalinity of ash"            "Potassium"
#[11] "Calcium"                      "Magnesium"
#[13] "Phosphate"                    "Chloride"
#[15] "Total phenols"                "Flavanoids"
#[17] "Nonflavanoid phenols"         "Proanthocyanins"
#[19] "Color intensity"              "Hue"
#[21] "OD280/OD315 of diluted wines" "OD280/OD315 of flavonoids"
#[23] "Glycerol"                     "2-3-butanediol"
#[25] "Total nitrogen"               "Proline"
#[27] "Methanol"

# PCA-cor whitened data
Z = whiten(forina1986$attrib, method="PCA-cor")

wt = as.integer(forina1986$type)
plot(Z[,1], Z[,2], xlab=expression(paste(Z[1])), ylab=expression(paste(Z[2])),
  main="Forina 1986 Wine Data", sub="PCA-cor Whitening", col=wt, pch=wt+14)
legend("topright", levels(forina1986$type)[1:3], col=1:3, pch=(1:3)+14 )
```

```
## relationship to UCI wine data

# UCI wine data is a subset
uciwine.attrib = forina1986$attrib[, c("Alcohol", "Malic acid", "Ash",
  "Alcalinity of ash", "Magnesium", "Total phenols", "Flavanoids",
  "Nonflavanoid phenols", "Proanthocyanins", "Color intensity", "Hue",
  "OD280/OD315 of diluted wines", "Proline")]

# two small differences compared to UCI wine data matrix
uciwine.attrib[172,"Color intensity"]  # 9.9 but 9.899999 in UCI matrix
uciwine.attrib[71,"Hue"] # 0.91 but 0.906 in UCI matrix
```

`lusc` *TCGA LUSC Data*

Description

A preprocessed sample of gene expression and methylation data as well as selected clinical covariates for 130 patients with lung squamous cell carcinoma (LUSC) as available from The Cancer Genome Atlas (TCGA) database (Kandoth et al. 2013).

Usage

```
data(lusc)
```

Format

`lusc$rnaseq2` is a 130 x 206 matrix containing the calibrated gene expression levels of 206 genes for 130 patients.

`lusc$methyl` is a 130 x 234 matrix containing the methylation levels of 234 probes for 130 patients.

`sex` is a vector recording the sex (male vs. female) of the 130 patients.

`packs` is the number of cigarette packs per year smoked by each patient.

`survivalTime` is number of days to last follow-up or the days to death.

`censoringStatus` is the vital status (0=alive, 1=dead).

Details

This data set is used to illustrate CCA-based data integration in Jendoubi and Strimmer (2019) and also described in Wan et al. (2016).

Source

The data were retrieved from TCGA (Kandoth et al. 2014) using the TCGA2STAT tool following the guidelines and the preprocessing steps detailed in Wan et al. (2016).

References

Jendoubi, T., Strimmer, K.: A whitening approach to probabilistic canonical correlation analysis for omics data integration. BMC Bioinformatics 20:15 <DOI:10.1186/s12859-018-2572-9>

Kandoth, C., McLellan, M.D., Vandin, F., Ye, K., Niu, B., Lu, C., Xie, M., andJ. F. McMichael, Q.Z., Wyczalkowski, M.A., Leiserson, M.D.M., Miller, C.A., Welch, J.S., Walter, M.J., Wendl, M.C., Ley, T.J., Wilson, R.K., Raphael, B.J., Ding, L.: Mutational landscape and significance across 12 major cancer types. Nature 502, 333–339 (2013). <DOI:10.1038/nature12634>

Wan, Y.-W., Allen, G.I., Liu, Z.: TCGA2STAT: simple TCGA data access for integrated statistical analysis in R. Bioinformatics 32, 952–954 (2016). <DOI:10.1093/bioinformatics/btv677>

Examples

```
# load whitening library
library("whitening")

# load TGCA LUSC data set
data(lusc)

names(lusc)
#"rnaseq2"         "methyl"          "sex"             "packs"
#"survivalTime"    "censoringStatus"

dim(lusc$rnaseq2) # 130 206 gene expression
dim(lusc$methyl)  # 130 234 methylation level

## Not run:
library("survival")
s = Surv(lusc$survivalTime, lusc$censoringStatus)
plot(survfit(s ~ lusc$sex), xlab = "Years", ylab = "Probability of survival", lty=c(2,1), lwd=2)
legend("topright", legend = c("male", "female"), lty =c(1,2), lwd=2)

## End(Not run)
```

---

nutrimouse *Nutrimouse Data*

Description

The nutrimouse dataset is a collection of gene expression and lipid measurements collected in a nutrigenomic study in the mouse studying 40 animals by Martin et al. (2007).

Usage

```
data(nutrimouse)
```

Format

A list containing the following components:

`gene` collects gene expression of 120 genes in liver tissue for 40 mice.

`lipid` collects concentrations of 21 lipids for 40 mice.

`diet` describes the diet of each mouse ("coc", "fish", "lin", "ref", or "sun").

`genotype` describes the genotype of each mouse: wild type ("wt") or PPARalpha deficient ("ppar").

Details

This data set is used to illustrate CCA-based data integration in Jendoubi and Strimmer (2019) and is also described in Gonzalez et al. (2008).

Source

The original data are available in the CCA R package by Gonzalez et al. (2008), see `nutrimouse`.

References

Gonzalez, I., Dejean, S., Martin, P.G.P, Baccini, A. CCA: an R package to extend canonical correlation analysis. J. Statist. Software 23:1–13 (2008)

Jendoubi, T., Strimmer, K.: A whitening approach to probabilistic canonical correlation analysis for omics data integration. BMC Bioinformatics 20:15 <DOI:10.1186/s12859-018-2572-9>

Martin, P.G.P., Guillou, H., Lasserre, F., Dejean, S., Lan, A., Pascussi, J.-M., Cristobal, M.S., Legrand, P., Besse, P., Pineau, T.: Novel aspects of PPARalpha-mediated regulation of lipid and xenobiotic metabolism revealed through a multigenomic study. Hepatology 54, 767–777 (2007) <DOI:10.1002/hep.21510>

Examples

```
# load whitening library
library("whitening")

# load nutrimouse data set
data(nutrimouse)

dim(nutrimouse$gene) # 40 120
dim(nutrimouse$lipid) # 40 21
levels( nutrimouse$diet ) #  "coc"  "fish" "lin"  "ref"  "sun"
levels( nutrimouse$genotype ) # "wt"   "ppar"
```

scca *Perform Canonical Correlation Analysis*

Description

scca computes canonical correlations and directions using a shrinkage estimate of the joint correlation matrix of X and Y.

cca computes canonical correlations and directions based on empirical correlations.

Usage

```
scca(X, Y, lambda.cor, scale=TRUE, verbose=TRUE)
cca(X, Y, scale=TRUE)
```

Arguments

X	First data matrix, with samples in rows and variables in columns.
Y	Second data matrix, with samples in rows and variables in columns.
lambda.cor	Shrinkage intensity for estimating the joint correlation matrix - see cor.shrink. If not specified this will be estimated from the data.
scale	Determines whether canonical directions are computed for standardized or raw data. Note that if data are not standardized the canonical directions contain the scale of the variables.
verbose	Report shrinkage intensities-

Details

The canonical directions in this function are scaled in such a way that they correspond to whitening matrices - see Jendoubi and Strimmer (2019) for details. Note that the sign convention for the canonical directions employed here allows purposely for both positive and negative canonical correlations.

The function scca uses some clever matrix algebra to avoid computation of full correlation matrices, and hence can be applied to high-dimensional data sets - see Jendoubi and Strimmer (2019) for details.

cca it is a shortcut for running scca with lambda.cor=0 and verbose=FALSE.

If scale=FALSE the standard deviations needed for the canonical directions are estimated by apply(X, 2, sd) and apply(X, 2, sd).

If X or Y contains only a single variable the correlation-adjusted cross-correlations K reduce to the CAR score (see carscore) described in Strimmer and Zuber (2011).

Value

scca and cca return a list with the following components:

K - the correlation-adjusted cross-correlations.

lambda - the canonical correlations.

WX and WY - the whitening matrices for X and Y, with canonical directions in the rows. If scale=FALSE then canonical directions include scale of the data, if scale=TRUE then only correlations are needed to compute the canonical directions.

PhiX and PhiY - the loadings for X and Y. If scale=TRUE then these are the correlation loadings, i.e. the correlations between the whitened variables and the original variables.

scale - whether data was standardized (if scale=FALSE then canonical directions include scale of the data).

lambda.cor - shrinkage intensity used for estimating the correlations (0 for empirical estimator)

lambda.cor.estimated - indicates whether shrinkage intenstiy was specified or estimated.

Author(s)

Korbinian Strimmer (http://strimmerlab.org) with Takoua Jendoubi.

References

Jendoubi, T., and K. Strimmer 2019. A whitening approach to probabilistic canonical correlation analysis for omics data integration. BMC Bioinformatics 20: 15. <DOI:10.1186/s12859-018-2572-9>

Zuber, V., and K. Strimmer. 2011. High-dimensional regression and variable selection using CAR scores. Statist. Appl. Genet. Mol. Biol. 10: 34. <DOI:10.2202/1544-6115.1730>

See Also

cancor and carscore.

Examples

```
# load whitening library
library("whitening")

# example data set
data(LifeCycleSavings)
X = as.matrix( LifeCycleSavings[, 2:3] )
Y = as.matrix( LifeCycleSavings[, -(2:3)] )
n = nrow(X)
colnames(X) # "pop15" "pop75"
colnames(Y) # "sr"   "dpi"  "ddpi"

# CCA

cca.out = cca(X, Y, scale=TRUE)
cca.out$lambda  # canonical correlations
cca.out$WX      # whitening matrix / canonical directions X
```

```
cca.out$WY      # whitening matrix / canonical directions Y
cca.out$K       # correlation-adjusted cross-correlations
cca.out$PhiX    # correlation loadings X
cca.out$PhiX    # correlation loadings Y

corplot(cca.out, X, Y)
loadplot(cca.out, 2)
# column sums of squared correlation loadings add to 1
colSums(cca.out$PhiX^2)

# CCA whitened data
CCAX = tcrossprod( scale(X), cca.out$WX )
CCAY = tcrossprod( scale(Y), cca.out$WY )
zapsmall(cov(CCAX))
zapsmall(cov(CCAY))
zapsmall(cov(CCAX,CCAY)) # canonical correlations


# compare with built-in function cancor
# note different signs in correlations and directions!
cancor.out = cancor(scale(X), scale(Y))
cancor.out$cor                  # canonical correlations
t(cancor.out$xcoef)*sqrt(n-1)   # canonical directions X
t(cancor.out$ycoef)*sqrt(n-1)   # canonical directions Y


## see "User guides, package vignettes and other documentation"
## for examples with high-dimensional data using the scca function
```

---

whiteningMatrix *Compute Whitening Matrix and Whiten Data*

Description

whiteningMatrix computes the whitening matrix W corresponding to the five natural whitening procedures discussed in Kessy, Lewin, and Strimmer (2018).

whiten whitens data X using the empirical covariance matrix $cov(X)$ as basis for computing the whitening transformation.

Usage

```
whiteningMatrix(Sigma, method=c("ZCA", "PCA", "Cholesky",
                                "ZCA-cor", "PCA-cor"))
whiten(X, center=FALSE, method=c("ZCA", "PCA", "Cholesky", "ZCA-cor", "PCA-cor"))
```

Arguments

Sigma	Covariance matrix.
method	Determines the type of whitening.
X	Data matrix, with samples in rows and variables in columns.
center	Center columns to mean zero.

Details

ZCA whitening, or Mahalanobis whitening ensures that the average covariance between whitened and orginal variables is maximal. Likewise, ZCA-cor whitening leads to whitened variables that are maximally correlated (on average) with the original variables.

In contrast, PCA and PCA-cor whitening lead to maximally compressed whitened variables, as measured by squared covariance and correlation, respectively.

Cholesky whitening is the unique whitening procedure that results from lower-triangular positive diagonal cross-covariance and cross-correlations matrices.

In PCA and PCA-cor eigenvector matrices with positive diagonal are used, in order to resolve the sign-ambiguity and also to make cross-correlations and cross-correlations positive diagonal.

For details see Kessy, Lewin, and Strimmer (2018).

ZCA-cor whitening is implicitely employed in computing CAT and CAR scores (cf. `catscore` and `carscore`).

Canonical correlation analysis (CCA) can also be understood as a special form form of whitening.

Value

`whiteningMatrix` returns a square whitening matrix `W`.

`whiten` returns the whitened data matrix $Z = XW'$.

Author(s)

Korbinian Strimmer (http://strimmerlab.org) with Agnan Kessy and Alex Lewin.

References

Kessy, A., A. Lewin, and K. Strimmer. 2018. Optimal whitening and decorrelation. The American Statistician. 72: 309-314. https://doi.org/10.1080/00031305.2016.1277159

See Also

`catscore` and `carscore`.

Examples

```
# load whitening library
library("whitening")

######

# example data set
# E. Anderson. 1935.  The irises of the Gaspe Peninsula.
# Bull. Am. Iris Soc. 59: 2--5
data("iris")
X = as.matrix(iris[,1:4])
d = ncol(X) # 4
n = nrow(X) # 150
colnames(X) # "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"

# estimate covariance
S = cov(X)

# ZCA-cor whitening matrix
W.ZCAcor = whiteningMatrix(S, method="ZCA-cor")

# whitened data
Z.ZCAcor.1 = tcrossprod(X, W.ZCAcor)
zapsmall( cov(Z.ZCAcor.1) )

# directly compute whitened data from X
Z.ZCAcor.2 = whiten(X, method="ZCA-cor")
zapsmall( cov(Z.ZCAcor.2) )
```

Index